INTELLIGENT AGENTS: TOOLS FOR THE COMMAND POST AND COMMANDER

LTC Michael Bowman, Dr. Gheorghe Tecuci, and Mihai Boicu

Introduction

You are a brigade commander, 72 hours into the fight, and running on caffeine and catnaps. Your operations officer is briefing you on possible courses of action (COAs) for a new mission. He recommends a COA and explains the advantages. Something about the COA bothers you, but you just can't put your finger on it. Fatigue, stress, or some other distraction is keeping you from recalling something that would make a difference in this decision. You make your best judgement and drive on, but your gut feeling leaves you thinking that there was a better way—if you had just had more time or a clearer head!

Our decisions are a function of our education, training, experience, and personal preference. There is ample evidence that the decisions we make under stress are generally not as good as those we make when we are well rested, comfortable, and relaxed. Thus, the U.S. military selects commanders based on their demonstrated ability to make good decisions under adverse conditions.

Revisit the opening scenario. The situation and the environment are the same, but this time you have another tool to assist you. An intelligent agent, trained by you to remember the lessons of a lifetime, will help you decide. The intelligent agent is software that runs on common computers and accesses data from your battle command systems and planning tools, regardless of whether they are powerful networked computer systems or handwritten notes and sketches.

The intelligent agent does not care how cold it is or how much sleep you have had. In seconds, it evaluates the COAs and presents you with a list of strengths, weaknesses, and issues for each of them. You quickly scan the list, discarding some and nodding agreement with others, until you come to the one or more gems that you recognize as being critical to the decision. Based on your own judgement and the recommendations of your staff, but now armed with a few additional key considerations, you make your decision.

These considerations might be based on planning guidelines you learned in a classroom, an after action review from an exercise you participated in, or on new enemy tactics. The intelligent agent combines the things a computer does best—sorting and sifting through data—with the things a human does best—learning from a lifetime of experience. It provides concise, relevant, and explainable considerations that commanders can take into account when making decisions. This is our vision for the use of intelligent agents in the command post of the near future!

Learning Agents Laboratory

Decision support and expert systems have been around for a while. To date, they have produced more hype than service, and they have played a very limited role in military systems. Even with today's rapid growth in computing power, most software products claiming to be intelligent don't solve complex, real-world problems.

The George Mason University (GMU) Learning Agents Laboratory (LALAB) is taking a novel approach to the creation and use of intelligent agents to solve complex problems. The goal of GMU's research is to develop methods and tools that allow users with minimal computer skills to easily build, teach, and maintain intelligent software agents.

GMU's initial research was part of the Defense Advanced Research Projects Agency (DARPA) High Performance Knowledge Base (HPKB) project. Additional support was provided by the U.S. Air Force Office of Scientific Research and the Army Battle Command Battle Laboratory. The work continues in the DARPA Rapid Knowledge Formation Project, still supported by the Air Force and now also supported by the Army War College. The goal of HPKB was to test the claim that with the latest artificial intelligence (AI) technology, large knowledge bases could be quickly built and updated. GMU-DARPA research indicates that with the right approach, intelligent agents can meet this goal.

Acquiring Knowledge

A major stumbling block in building intelligent systems that solve problems equal to a human subject matter expert (SME) is the "knowledge acquisition bottleneck." This bottleneck comes from the requirement to transfer knowledge from an expert, through a knowledge engineer, to the computer. The knowledge engineer must learn what the expert knows and how the expert uses that knowledge. The engineer then uses various tools and techniques to build a knowledge base. This is a long, painful, and inefficient process.

The GMU approach, called "Disciple," is a theory, methodology, and tool set in which an SME directly constructs an intelligent agent. In this approach, SMEs teach the agent to perform various tasks in a way that resembles how they would teach an apprentice or student. They give the agent examples and explanations, and supervise and correct its behavior.

The traditional approach to create a useful knowledge base requires very complex steps, including the creation of an ontology that defines relevant terms and relationships from a problem domain, the definition of problem-solving rules, and the validation and update of these rules. In general, these tasks require the creation of formal computer representations, a task that only a knowledge engineer can accomplish.

In the Disciple approach, complex tasks are replaced with simpler ones. Instead of creating an ontology, the expert updates and extends an initial ontology imported from existing sources of knowledge. Further, instead of defining a complex problem-solving rule, the expert identifies and explains an example solution from which Disciple learns a general rule. In lieu of debugging a complex problem-solving rule, the expert critiques specific examples of agent problem solving from which Disciple updates corresponding rules.

The expert will not need to create formal computer representations, just understand information generated by Disciple. Finally, the expert will not need to provide formal explanations, just informal hints that will guide Disciple in generating possible explanations from which the expert will choose.

Disciple's history, capabilities, and inner workings are described in detail in Dr. Gheorghe Tecuci's *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies,* Academic Press, 1998. Recent papers describing improved capabilities are also available on the GMU LALAB Web page at **http://lalab.gmu.edu**.

A Sample Application

As part of HPKB, the GMU LALAB developed a Disciple agent to critique COAs for ground-combat operations. The COAs were provided by the Army and came in a standard format consisting of a multiparagraph description and a tactical sketch.

The Disciple COA agent identifies strengths and weaknesses of a COA with respect to the principles of war and the tenets of Army operations as described in Army Field Manual 100-5. A general understanding of the principles and tenets exists, but military experts disagree on their application. The GMU LALAB's goal was to create a tool that contained this common understanding while being flexible enough to allow rapid personalization by the SME training and using the agent. The following is an example of a strength identified by Disciple in a COA for the principle of surprise:

"There is a strength with respect to surprise in COA411 because the enemy is unlikely to be prepared for the heavy concentration of combat power applied by the Blue-force main-effort during its penetration. In this action, the main-effort is applying a force ratio of 10.6, which is more than double the recommended force ratio of 3.0. Applying this much combat power for this penetration is likely to surprise the enemy and is indicative of the proper application of the principle of surprise."

Building Disciple Agents

The development of a specific Disciple agent includes two main processes: ontology development and agent training. Building the domain ontology begins with importing background military knowledge such as unit echelons and capabilities from existing sources of knowledge. Additional terms and relationships identified by the expert are added as necessary. The Disciple-COA ontology was built by importing many terms needed to model the COA domain from a research knowledge base developed by Cycorp, called CYC.

Training a Disciple agent is an iterative process of showing it how to solve problems based on examples, letting the agent attempt to solve other problems, and providing the agent explanations for why these solutions are right or wrong. A strength of this approach is that the expert does not have to be perfect or comprehensive when conducting agent training. Flaws in training show up naturally when Disciple tries to solve problems on its own. The expert merely has to examine Disciple's solutions and provide explanations regarding where it went wrong.

Experimental Results

The Disciple methodology and agents have been tested with other systems as part of DARPA annual HPKB Program evaluations. In summary, the experimental results show that Disciple-based agents were highly effective in knowledge acquisition and complex problem solving, outperforming other systems developed to solve similar problems.

In August 1999, the GMU LALAB conducted a knowledge-acquisition experiment to demonstrate that it is possible for military experts to directly train Disciple agents. Four Army officers successfully trained Disciple agents that critiqued COAs. Commenting on the usefulness of Disciple, LTC John N. Duquette stated, "The potential use of this tool by domain experts is only limited by their imagination—not their AI programming skills." We believe this is the first time SMEs with no prior knowledge or engineering experience successfully trained intelligent agents to solve complex problems.

Conclusion

This article briefly presented a vision for using intelligent agents in a military command post, described some of the challenges, and presented the Disciple approach to overcoming those challenges. The long-term goal of the GMU LALAB is to develop technology that allows typical computer users to directly build intelligent agents and knowledge bases as easily as they use personal computers for text processing. This will change the way intelligent agents are built, from being programmed by a knowledge engineer to being taught by an SME, and will contribute to a generalized application of agent technology in all areas of human activity.

LTC MICHAEL BOWMAN was a student at the Army War College and a Ph.D. candidate at GMU at the time he wrote this article. He received a B.S. degree from Ouachita Baptist University and an M.S. degree from the Naval Postgraduate School.

DR. GHEORGHE TECUCI is Professor of Computer Science, Director of the GMU LALAB, and a member of the Romanian Academy. He received M.S. and Ph.D. degrees in computer science from the Polytechnic University of Bucharest and another Ph.D. degree in computer science from the University of Paris-South. He has published more than 100 scientific papers and 5 books, most of them in artificial intelligence.

MIHAI BOICU was a Ph.D. student in computer science at GMU and a Graduate Research Assistant in the GMU LALAB at the time he wrote this article. He has published more than 15 papers and is a member of the American Association for Artificial Intelligence.